Decision making with limited feedback: Error bounds for predictive policing and recidivism prediction *

Danielle Ensign DANIENSIGN@GMAIL.COM
School of Computing, University of Utah

Sorelle A. Friedler SORELLE@CS.HAVERFORD.EDU
Haverford College

Scott Neville DROP.SCOTT.N@GMAIL.COM
School of Computing, University of Utah

Carlos Scheidegger CSCHEID@CSCHEID.NET
University of Arizona

Suresh Venkatasubramanian SURESH@CS.UTAH.EDU
School of Computing, University of Utah

Editors: Mehryar Mohri and Karthik Sridharan

Abstract

When models are trained for deployment in decision-making in various real-world settings, they are typically trained in batch mode. Historical data is used to train and validate the models prior to deployment. However, in many settings, *feedback* changes the nature of the training process. Either the learner does not get full feedback on its actions, or the decisions made by the trained model influence what future training data it will see.

In this paper, we focus on the problems of recidivism prediction and predictive policing. We present the first algorithms with provable regret for these problems, by showing that both problems (and others like these) can be abstracted into a general reinforcement learning framework called partial monitoring. We also discuss the policy implications of these solutions.

Keywords: Partial monitoring, online learning, predictive policing, recidivism prediction

1. Introduction

Machine learning models are increasingly being used to make real-world decisions such as who to hire, who should receive a loan, where to send police, and who should receive parole. These deployed models mostly use traditional batch-mode machine learning, where decisions are made and observed results supplement the training data for the next batch.

However, the problem of *feedback* makes traditional batch learning frameworks both inappropriate and incorrect. Hiring algorithms only receive feedback on people who were hired, and predictive policing algorithms only observe crime in neighborhoods they patrol. Secondly, decisions made by the system influence the data that is fed to it in the future. For example, once a decision has been made to patrol a certain neighborhood, crime from *that* neighborhood will be fed into the training apparatus for the next round of decision-making.

In this paper, we model these problems in a reinforcement learning setting, and derive algorithms with provable error bounds. Notably, these algorithms also translate into concrete procedures that differ from current practice in the problems under study.

* This research is funded in parts by the NSF under grants IIS-1633387, IIS-1633724 and IIS-1513651.

The problems We will focus on two problems that are of particular societal importance: predictive policing and recidivism prediction. These problems are at the core of the algorithmic pipeline in criminal justice through which automated decision-making has a material impact on society. They also serve as archetypal problems through which we can gain an understanding of generalizable issues faced in deployment. Another motivating factor is that systems for solving these problems are already in use and issues with these processes are already documented, making the discussion of remedies urgent. While problems with recidivism prediction have been documented in the well-publicized and Pulitzer-prize finalist work by ProPublica (Angwin et al., 2016), the complications that arise from limited feedback have not been discussed. PredPol, a predictive policing system, has been shown to produce inaccurate feedback loops when deployed in batch mode (Lum and Isaac, 2016), so that police are repeatedly sent back to the same neighborhoods, even though the underlying crime rate would suggest a different deployment.

Definition 1 (Predictive Policing) *Given historical crime data for a collection of d regions, decide how to allocate k patrol officers to areas to detect crime.*

Definition 2 (Recidivism Prediction) *Given an inmate up for parole, use a model of* re-offense *(whether the individual will reoffend within a fixed time period after being released) to determine whether they should be granted parole.*

Contributions. Our first contribution is a formal model for predictive policing which places it in the framework of partial monitoring. We exploit structure within the problem to reduce it to a combinatorial *semi-bandit* problem. Our reduction, combined with existing regret bounds for such problems, yields an algorithm (the first of its kind) for predictive policing that exhibits $O(kd\sqrt{kT})$ regret over T iterations. This result, and the method used to prove it, is somewhat counter-intuitive: the "true loss" i.e the actual crime rate is not revealed to the learner, but we show that there are fully observable *proxy losses* that yield the same instantaneous (and therefore overall) regret.

We also consider the degree to which feedback affects algorithm performance, by considering instead the case when crime is *reported* instead of *discovered* by patrol officers. Using our framework from above, we show that this can be analyzed using a full information online linear optimization framework, yielding an algorithm with regret $O(kd\sqrt{T \log k})$.

Turning now to recidivism prediction, we show that it too has a natural analog in the partial monitoring literature, in the form of the *apple tasting* problem. By invoking results in that model, we present an algorithm (the first with a provable guarantee) for recidivism prediction that achieves a mistake bound of $\sqrt{T}$.

We also examine the policy implications of these results. In the case of predictive policing, our results provide an alternative to current deployed algorithms that are based on batch learning and are vulnerable to runaway feedback loops (Ensign et al., 2017). In the case of recidivism prediction, our algorithm suggests a random process by which inmates are released: while this might not be a tenable practical solution, it resembles closely practical approaches involving the random assignment of judges to decision-making.

2. Related Work

Our work fits into the larger framework of the social implications of algorithmic decision-making, and as such it overlaps with the recent interest in fairness, accountability, and transparency of these systems. The narrower question of defining notions of fairness in *sequential* learning settings such as the ones we describe has been studied extensively, primarily in the setting of bandits (regular, contextual and linear) and Markov decision processes (Kannan et al., 2017; Joseph et al., 2016b; Jabbari et al., 2016; Joseph et al., 2016a). There, the primary goal is to understand how to define fairness in such a process, and how ensuring fairness might affect the ability to learn an accurate model.

We note that the perspective from Markov decision processes (and POMDPs) has much to offer: however, the problems of limited feedback relate more directly to the area of *partial monitoring* (Cesa-Bianchi and Lugosi, 2006) which we employ heavily in this paper. There are a number of systems currently in place for recidivism predic-

tion and predictive policing. While the details of the actual implementations (such as COMPAS (NorthPointe, Inc.)) remain proprietary, Berk and Bleich (2013) provide a comprehensive review of the methods used in this area. There has been important *empirical* work (Lum and Isaac, 2016) demonstrating the consequences of feedback loops in simulation in the predictive policing setting (specifically the system known as PredPol (Mohler et al., 2015)).

3. Background

The reinforcement learning framework we will be using to evaluate the above problems is the well-known *partial monitoring* framework (Piccolboni and Schindelhauer, 2001),(Cesa-Bianchi and Lugosi, 2006, Chapter 6). Formally, a partial monitoring problem $P = (A, Y, H, L)$ consists of a set of n actions $A = \{a_1, a_2, ..., a_n\}$ and a set of m outcomes (adversary actions) $Y = \{y_1, y_2, ..., y_m\}$. There is a feedback function (also called a feedback matrix) $H\colon A \times Y \to \Sigma$ that takes in a learner action and an outcome and outputs some symbol $\sigma \in \Sigma$ denoting information that the learner receives. Finally there is a loss function (also called a loss matrix) $L\colon A \times Y \to \mathbb{R}$ that takes in an action and an outcome and outputs a loss (which is usually assumed to be positive). We denote $h(a_t, y_t) \in \Sigma$ as the value of H given an action and an outcome, and $\ell(a_t, y_t) \in \mathbb{R}$ as the value of L given an action and an outcome. The learner and adversary are told what L and H are before the learning begins. After the learner performs a given action, they do *not* have access to the incurred loss; in partial monitoring, the loss is said to be *hidden*.

As usual, an algorithm consists of sequence of actions, and its quality is measured in terms of regret bounds, (either weak, strong or stochastic). Standard multi-arm bandits (Bubeck et al., 2012) can be captured in this setting by setting the feedback matrix H to be equal to the loss matrix L.

In general, proving regret bounds for partial monitoring is hard because the feedback matrix H might bear no relation to the true loss matrix L. Thus, results in this area take two forms. One class of results look at general bounds on partial monitoring under assumptions about the relation between H and L(Bartók et al., 2014) and another class of results look at special subcases that are more amenable to analysis (such as the vast literature on bandits(Bubeck et al., 2012)).

Regret and Mistake Bounds For any partial monitoring algorithm, let the algorithm actions be $a_1, a_2, \ldots, a_T$ with corresponding outcomes $o_1, o_2, \ldots, o_T$. Note that the actions might be random variables. Then the (weak) regret of the algorithm is its loss compared to the loss of any fixed action:

$$R_T = \sum_{i \in T} \ell(a_i, o_i) - \min_{a \in A} \sum_{t \leq T} \ell(a, o_i)$$

and the *expected* weak regret is $E[R_T]$. Our goal will be to optimize this quantity in a minimax way, (i.e over all adversaries and all strategies).

Alternately, we can measure algorithm performance in terms of *mistake bounds*. A mistake is an action-outcome pair for which $\ell(a, o) > 0$, and the mistake bound of an algorithm is the number of mistakes. Note that mistake bounds are not relative with respect to some fixed action.

4. Modeling Predictive Policing

We now formalize predictive policing in a partial monitoring setting. Assume we have a police force consisting of k officers patrolling a set of d regions. An *action* $\mathbf{a} = (a_1, a_2, \ldots, a_d), 0 \leq a_i \leq k, \sum_i a_i = k$ consists of a deployment of the k officers to the d regions: formally, the set $A = \{(a_1, a_2, \ldots, a_d) \mid 0 \leq a_i \leq k, \sum_i a_i = k\}$ consists of all possible ordered partitions of d into k parts and has size $|A| = O((\frac{d+k}{k})^k)$. An *outcome* describes the actual amount of crime in each of the regions and is expressed as the tuple $\mathbf{o} = (o_1, o_2, \ldots, o_d), o_i \geq 0$, and the set of all possible outcomes is denoted by O. We will assume that the total crime $\sum_i o_i$ on any day is finite but unbounded. However we will show that from the perspective of algorithm design without loss of generality we can assume that the total crime is upper bounded by kd.

For predictive policing the (hidden) loss is the number of uncaught crimes. We will make the simplifying modeling assumption that each officer in a region can "catch" one crime: while this does not reflect the reality of police patrolling, abstractly, this can be thought of as a measure of

how much work an officer can do in a given area in one time step. Too many officers in one place will create slack, while too few officers leads to missed opportunity. The linearity assumption captures the idea that each productive officer is roughly equivalent (at least at the limit). We will use the analogy to crime presence and catching crime throughout. With that assumption in place, the system incurs loss when a region does not have sufficiently many officers assigned to catch all the crime in it. Thus, given vectors for action $\mathbf{a}$ and outcome $\mathbf{o}$, we define the total loss as

$$L(\mathbf{a}, \mathbf{o}) = \sum_i \max(o_i - a_i, 0) \tag{1}$$

Officers that catch crime relay this information back to their precinct. Thus, the *feedback* received by the system is merely the amount of caught crime. For any region i this can be expressed as $h(a_i, o_i) = \min(a_i, o_i)$, and so the *feedback vector* received by the system is

$$H(\mathbf{a}, \mathbf{o}) = (h(a_1, o_1), \ldots, h(a_d, o_d)) \tag{2}$$

Our partial monitoring system for predictive policing can now be described as the system (A, O, L, H).

5. An Observable Form of Predictive Policing

The system (A, O, L, H) is a classic *partial* monitoring system because it is not obvious how to relate the loss and feedback matrices. Using a structural characterization developed by Neu and Bartók (2013), it is possible to show a bound of $O(T^{2/3})$ by exploiting useful symmetries in L and H. However, we can do better. Through a series of reduction, we now show that a proxy loss function achieves the same weak regret as L but has the advantage of being fully observable given H, thus yielding a more traditional bandit formulation.

Recall that our original loss function L is given by Equation (1). We now define two different loss functions that are related to L. As always, let $\mathbf{a}$ be the vector of actions (i.e the number of officers assigned to each region) and let $\mathbf{o}$ be the vector of *outcomes*: the amount of crime that occurred in each region.

Loss: The number of officers who failed to catch crime. If we send too many officers to any region, then we run the risk of over-provisioning: because the total number of officers is fixed, this might result in too few officers sent elsewhere. A loss based the number of officers who *failed* to catch crime is

$$L_f(\mathbf{a}, \mathbf{o}) = \sum_i \max(0, a_i - o_i) \tag{3}$$

Note that while L is not directly observable from the feedback H, L_f is observable since in each region the feedback indicates whether $o_i \geq a_i$ and if not yields o_i from which we can compute $a_i - o_i$.

Loss: The number of wasted officers. The above loss is misleading: if the total crime overwhelms the number of available officers, then there is really nothing the officers can do. A more accurate measure might be the number of officers *who could have caught crime* but did not (because they were misallocated). This involves a slight correction to the previous loss: let C^t denote the total crime at a given time instant. Then

$$L_w(\mathbf{a}, \mathbf{o}) = -\max(0, k - C^t) + \sum_i \max(0, a_i - o_i) \tag{4}$$

Note that the only difference between $L_w(a^t, c^t)$ and $L_f(a^t, c^t)$ is the term $-max(0, k - C^t)$. This difference means that L_w is not directly observable from feedback (since we don't know C^t).

Proving equivalence Our goal is now to show that from the perspective of optimizing (strong or weak) regret, L (true but unobserved) is equivalent to L_f (observable). We'll do this by showing that the regret function doesn't change when we change our loss. This is unusual, as L_f is clearly observable, yet L, L_w are not.

Let $\mathbf{a}^t$ denote the vector of actions taken at time t and similarly let $\mathbf{o}^t$ be the outcome vector at time t. Let C^t denote the total crime occurring at time t. In either form of regret, we compare our loss at time t to the best action (at time t for strong regret, and the single best for weak regret). Let this action be $\mathbf{b}^t$. At time t, the instantaneous regret from an action $\mathbf{a}^t$ with respect to loss ℓ is given by

$$R(\mathbf{a}^t, \mathbf{b}^t, \ell) = \ell(\mathbf{b}^t, \mathbf{o}^t) - \ell(\mathbf{a}^t - \mathbf{o}^t)$$

We now prove our equivalence in two steps.

Lemma 3

$$R(\mathbf{a}^t, \mathbf{b}^t, L_w) = R(\mathbf{a}^t, \mathbf{b}^t, L_f)$$

Proof

Let us first consider the instantaneous regret with respect to L_w (Equation (4)).

$$\begin{aligned} R(\mathbf{a}^t, \mathbf{b}^t, L_w) &= L_w(\mathbf{b}^t) - L_w(\mathbf{a}^t) \\ &= \Big(- \max(0, k - C^t) \\ &\quad + \sum_{i=1}^{n} \max(0, b_i^t - o_i^t) \Big) \\ &\quad - \Big(- \max(0, k - C^t) \\ &\quad + \sum_{i=1}^{n} \max(0, a_i^t - o_i^t) \Big) \end{aligned}$$

which after cancelling the max terms yields

$$\Big(\sum_{i=1}^{n} max(0, b_i^t - c_i^t) \Big) - \Big(\sum_{i=1}^{n} \max(0, a_i^t - c_i^t) \Big)$$

If we now consider L_f, the instantaneous regret is simply

$$\begin{aligned} R(\mathbf{a}^t, \mathbf{b}^t, L_w) &= \Big(\sum_{i=1}^{n} max(0, b_i^t - c_i^t) \Big) \\ &\quad - \Big(\sum_{i=1}^{n} \max(0, a_i^t - c_i^t) \Big) \end{aligned}$$

which means that at any time t and for any action $\mathbf{b}$

$$R(\mathbf{a}^t, \mathbf{b}^t, L_w) = R(\mathbf{a}^t, \mathbf{b}^t, L_f)$$

■

The next step is to connect the true loss L to the auxiliary loss L_w.

Lemma 4

$$R(\mathbf{a}^t, \mathbf{b}^t, L) = R(\mathbf{a}^t, \mathbf{b}^t, L_w)$$

Proof

Fix a time t. If $C^t \leq k$, $L(\mathbf{a}^t, \mathbf{o}^t) = L_w(\mathbf{a}^t, \mathbf{o}^t)$ directly from (1),(4). Intuitively, this is because we have enough officers to catch all crime, and so crime not caught corresponds to wasted allocation. Conversely, If $C^t > k$, we are guaranteed to miss $C^t - k$ crimes regardless of how we assign officers. Thus, when $C^t > k$, we get

$$L_w(\mathbf{a}^t, \mathbf{o}^t) = L(\mathbf{a}^t, \mathbf{o}^t) - (C^t - k)$$

We can summarize these relationships as

$$L_w(\mathbf{a}^t, \mathbf{o}^t) = L(\mathbf{a}^t, \mathbf{o}^t) - \max(0, C^t - k)$$

Because this is a constant independent of the action taken at time t, the same logic as in Lemma 3 can be applied to conclude that

$$R(\mathbf{a}^t, \mathbf{b}^t, L_w) = R(\mathbf{a}^t, \mathbf{b}^t, L)$$

■

Combining Lemmas 3 and 4 shows that regret with respect to the true loss L is equivalent to regret with respect to the observable loss L_f. This means that in the regret sense, the best strategy for one Loss function gives the best strategy for the other.

6. An Algorithm for Predictive Policing

The results of the previous section allow us to cast predictive policing as a classical multi-armed bandit, where each "arm" is a particular allocation of officers to regions. Using standard regret bounds for multi-armed bandits, this implies an algorithm that in T steps incurs $O(\sqrt{TK})$ regret (Auer et al., 1995) where K is the number of arms.

However, the number of arms K is prohibitive: it is the number of ways of partitioning the integer k into d parts, and as such is exponential in d. A better way to model this problem is to exploit the vector structure of the actions and losses, in the form of a *semi-bandit* problem.

Definition 5 (Semi-bandits (Neu and Bartók, 2013)) *In each time step the learner chooses an* action $\mathbf{V}_t$ *drawn from a set* $S \subseteq \{0,1\}^d$*, where for all* $\mathbf{v} \in S, \|v\|_1 \leq m$*. The environment picks a loss vector* $\boldsymbol{\ell}_t \in [0,1]^d$*, and the learner incurs a loss* $\mathbf{V}_t^\top \boldsymbol{\ell}_t$ *while receiving the feedback vector* $(V_{t,1}\ell_{t,1}, \ldots, V_{t,d}\ell_{t,d})$*.*

Semi-bandit feedback provides more feedback than a classical bandit (in which we would merely get the feedback $\mathbf{V}_t^\top \boldsymbol{\ell}_t$, while not being as powerful as a full information setting (where we would receive the environment vector $\boldsymbol{\ell}_t$). The key value of the semi-bandit formulation is that one can design algorithms with regret $O(\sqrt{mdT})$ (Audibert et al., 2013).

6.1. Mapping predictive policing to semi-bandits

We now describe how to transform the bandit problem (A, O, L_f, H) into a semi-bandit problem. Consider the function $f_o : [0 \dots k] \to [0 \dots k]$ defined as $f_o(x) = \max(0, x - o)$. We associate with f_o the $k+1$-dimensional vector $v_o = (f_o(0), f_0(1), \dots, f_o(K))$. Finally, given an outcome vector $\mathbf{o}$, we construct the environment vector $\boldsymbol{\ell}_\mathbf{o} = (v_{o_1}, v_{o_2}, \dots, v_{o_d})$.

We now turn to the encoding of actions. For any $0 \le a \le k$, let u_a be the unit vector of dimension $k+1$ with a 1 in position a. We can now encode the action $\mathbf{a}$ as the vector $V_\mathbf{a} = (u_{a_1}, u_{a_2}, \dots, u_{a_d})$. The loss $V_\mathbf{a} \cdot \boldsymbol{\ell}_\mathbf{o}$ is $\sum_{i=1}^d \max(0, a_i - o_i) = L_f(\mathbf{a}, \mathbf{o})$.

An Illustration. For concreteness, consider a setting with $d = 3$ regions and $k = 4$ officers. Suppose that at time t, the actual crime vector is $o_t = (2, 1, 4)$. This corresponds to an environment loss $\boldsymbol{\ell}_t$ of the form (here written stacked)

$$\boldsymbol{\ell}_t = \begin{matrix} 0 & 0 & 0 & 1 & 2 & \dots v_{o_1} \\ 0 & 0 & 1 & 2 & 3 & \dots v_{o_2} \\ 0 & 0 & 0 & 0 & 0 & \dots v_{o_3} \end{matrix}$$

Suppose the algorithm now chooses the action $a_t = (1, 3, 0)$. Then:

$$V_t = \begin{matrix} 0 & 1 & 0 & 0 & 0 & \dots u_{a_1} \\ 0 & 0 & 0 & 1 & 0 & \dots u_{a_2} \\ 1 & 0 & 0 & 0 & 0 & \dots u_{a_3} \end{matrix}$$

Thus $V_t \cdot l_t = 0 + 2 + 0 = 2$ as desired since we have 2 officers who did not catch a crime.

The dimension of $V_\mathbf{a}$ is $d \cdot k$. Further, note that $|V_\mathbf{a}|_1 = d$. To apply the bound above, we need V_t, l_t to be $\{0, 1\}$, which we can arrange by encoding all values in unary and replacing each dimension with k entries. So that $(.., 3, ..)$ becomes $(..., 1, 1, 1, 0, ...)$ in l_t. The result is that our unary vectors are a factor of k larger, and $|V_t| = k \cdot d$. Substituting these terms into the bound from above, we obtain the following result.

Theorem 6 *There is an algorithm for predictive policing that in T steps incurs total (minimax) regret $O(kd\sqrt{Tk})$.*

Note we have eliminated the dependence on the (large) number of arms, in favor of a new $k^{\frac{3}{2}}d$ term.

6.2. Reported versus Discovered Crime

Thus far, we have only considered one kind of crime "incident": a crime discovered by a patrolling officer. In general, incidents might also be recorded via reports from residents in a region. We call the former incident *discovered* crime and the latter incident a *reported* crime.

We will assume that reported crime is not affected by feedback (i.e that the likelihood of calling 911 is independent of the current level of policing in an area. While this might not always be an accurate model of what prompts crime reports, it allows us to understand how the presence of reported incidents might alleviate the problem of (limited) feedback.

In the most general case, *all* crime is reported. In our model, this is equivalent to providing $\mathbf{o}_t$ as the feedback in time t. This reduces to a traditional online linear optimization in the "vector" feedback setting from Section 6.1 above. Employing the same reduction, and using known results for online linear optimization due to (Audibert et al., 2013), we can conclude the following.

Theorem 7 *If all crime is reported, there is an algorithm for predictive policing that incurs regret $O(kd\sqrt{T \log k})$*

Notice that eliminating feedback reduces the dependence on the number of officers, but does not affect the dependence on the number of regions. We leave as an open question the best bound attainable if we receive some fixed (but unknown) fraction of the reported incidents in addition to the crime discovered through policing.

7. Recidivism Prediction

We now formalize the problem of recidivism prediction in the context of partial monitoring. Recall from Section 1 that recidivism prediction is the problem of determining if someone convicted of a crime will reoffend if released (often measured

based on rearrest within a fixed time period, say, 2 years). Such predictions are then used to determine if parole should be granted. The *action* here is the decision to grant parole, and the *outcome* is whether a crime is subsequently committed or not. Formally, we will assume two actions, `keep` and `release` and two outcomes, `c` ("crime") and ¬`c` ("no crime"). We can then define a *feedback* matrix and a corresponding loss matrix

$$L = \begin{array}{c} \\ \texttt{keep} \\ \texttt{release} \end{array} \begin{array}{c} \begin{array}{cc} \texttt{c} & \neg\texttt{c} \end{array} \\ \begin{pmatrix} 0 & b \\ c & d \end{pmatrix} \end{array}$$

$$H = \begin{array}{c} \\ \texttt{keep} \\ \texttt{release} \end{array} \begin{array}{c} \begin{array}{cc} \texttt{c} & \neg\texttt{c} \end{array} \\ \begin{pmatrix} - & - \\ c & d \end{pmatrix} \end{array}$$

In what follows, we will assume that $c = b = 1$ and $d = 0$. However in general, one might assign different values to b, c and d if one had different risk associated with incorrect release (c) versus an unfair (d) or valid (b) incarceration. In recidivism prediction, *context* consists of profile information about the individual being evaluated.

7.1. A connection to apple tasting

Apple tasting is a well known example of partial monitoring that can be solved with good regret bounds. In the apple tasting problem, the goal is to test apples in an orchard prior to selling them. If we test an apple by tasting it, we discover if it is bad or good, (but we cannot then sell it and incur a loss if it was good). If we sell the apple, then we receive a loss if it is bad and no loss if it is good. In this setting the *partial* monitoring comes from the algorithm only receiving feedback if it decides to taste. However, the algorithm incurs a hidden loss if it tastes a good apple or sells a bad one.

Formally, we can encode this as partial monitoring with the following loss and feedback matrices:

$$L = \begin{array}{c} \\ \texttt{sell} \\ \texttt{taste} \end{array} \begin{array}{c} \begin{array}{cc} \texttt{bad} & \texttt{good} \end{array} \\ \begin{pmatrix} 0 & b \\ c & d \end{pmatrix} \end{array}$$

$$H = \begin{array}{c} \\ \texttt{sell} \\ \texttt{taste} \end{array} \begin{array}{c} \begin{array}{cc} \texttt{bad} & \texttt{good} \end{array} \\ \begin{pmatrix} - & - \\ c & d \end{pmatrix} \end{array},$$

The *context* here is provided by the description of the apple: its color, texture and so on. The key observation we make here is that: *apple tasting is equivalent to recidivism prediction.* This leads us to a regret bound for recidivism prediction.

Lemma 8 (via Antos et al. (2013)) *There exists a minimax $O(\sqrt{T})$ weak regret algorithm for recidivism prediction where T is the number of time steps, and this can be achieved using the* EXP3 *algorithm of Auer et al. (2002)*

7.1.1. Mistake Bounds

The particular structure of the apple tasting problem allows for a stronger analysis. Helmbold et al. (2000b) presented algorithm that achieves a mistake bound of $\sqrt{T}$ for apple tasting.Moreover, their bounds apply even when we have *context*. As before, this immediately yields a similar bound for recidivism prediction.

To state the result we must first assume the existence of an online binary classifier that makes a total of M_+ false positive and M_- false negative errors in T steps. As Helmbold et al. (2000b) show, such a classifier can be obtained from related results (Helmbold et al., 2000a). The bound for recidivism prediction can then be stated as follows.

Lemma 9 (via Helmbold et al. (2000b)) *There exists an algorithm for recidivism prediction whose expected mistake bound upper bounded by $M_+ + 2\sqrt{TM_-}$.*

Algorithms The results above come with algorithms that achieve the desired error bounds. In the interest of space we focus on the stronger result from Lemma 9. The key insight here (and in many such methods) is that in order to defeat the adversary, the algorithm must decide *at random* when to request an evaluation (i.e release an inmate). More formally, the algorithm asks the online classifier to make a prediction. If the classifier predicts ¬`c`, the recidivism predictor does the same. If not, the predictor tosses a coin and with probability roughly $\sqrt{M/T}$ decides to release the inmate anyway and obtains feedback. Over time, the probability of overriding the classifier decreases (as its accuracy increases). We refer the reader to Helmbold et al. (2000b) for details of the proof.

Policy Implications The algorithm presented above suggests that inmates should be released *at random* in order to build an effective model for recidivism prediction. The practical remedy for this problem has been to observe that judges are assigned to cases in a random way. Suppose each judge j is modeled by a predictor $p_j(x)\colon \mathcal{X} \to [0,1]$ that takes a personal profile x and releases the person with probability $p_j(x)$. If cases are assigned uniformly at random to one of k judges then the probability of an individual being released is $\frac{1}{k}\sum_j p_j(x)$. As long as individual judge bias (captured in $p_j(x)$) is distributed across the range, this effectively corresponds to releasing an individual uniformly at random.

The claim that current methodology captures the random labeling recommended by Lemma 9 rests on a key assumption: for any input x, $\frac{1}{k}\sum_j p_j(x)$ is close to $1/2$. Once we stratify by crime, this might not be true at all: for example, most judges may be less likely to grant convicted murderers parole. In this case, $\frac{1}{k}\sum_j p_j(x) \ll 1/2$. We leave for future research the question of how stable the $\sqrt{T}$-mistake bound algorithm is under such weaker definitions of randomness.

8. Discussion

The results presented here are the first time the problem of feedback in automated decision-making has been framed in the (arguably natural) online learning setting. Existing tools for these problems are typically based on traditional batch learning frameworks, with no rigorous approach updating the models thus learned. It is therefore reasonable to ask: should all such algorithms be replaced with methods based on our results?

Consider first predictive policing. Our bounds for regret are predicated on the idea that minimizing uncaught crime is the goal. However, an alternate goal is the more modest "learn the crime rates in different regions" (this is in fact how systems like PredPol are designed). That framing corresponds to a stochastic model for the environment rather than the adversarial model we have adopted here. Further, one might then argue that considering *strong* regret might be more appropriate. It remains an interesting open question to consider the semi-bandit problem we construct in this setting. While the work of (Audibert et al., 2013) and (Neu and Bartók, 2013) addresses stochastic environments, they do not address the setting of strong regret. We note here that our general reduction holds for instantaneous regret and therefore our equivalence between losses applies even for strong regret. Note that a natural extension of this model would be to introduce context, which might allow us to leverage the literature on contextual semi-bandit learning (see (Krishnamurthy et al., 2016) and references therein). Interestingly, context is explicitly avoided in PredPol because of the fear of biased decision-making.

In the case of recidivism prediction, the simplified version we presented results in a binary classification of inmates. However, in general such a prediction system typically outputs a risk level – a score between 1 and 10 for example – and so might fall into a more general setting where the feedback depends on the score in a more complicated manner. Modeling such a scenario would be of great practical interest.

References

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica*, May 23, 2016.

András Antos, Gábor Bartók, Dávid Pál, and Csaba Szepesvári. Toward a classification of finite partial-monitoring games. *Theoretical Computer Science*, 473:77–99, 2013.

Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2013.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Gábor Bartók, Dean P Foster, Dávid Pál, Alexander Rakhlin, and Csaba Szepesvári. Partial monitoring—classification, regret bounds, and

algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

Richard A. Berk and Justin Bleich. Statistical procedures for forecasting criminal behavior. *Criminology & Public Policy*, 12(3):513–544, 2013. ISSN 1745-9133. doi: 10.1111/1745-9133.12047. URL http://dx.doi.org/10.1111/1745-9133.12047.

Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Danielle Ensign, Sorelle A Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. Runaway feedback loops in predictive policing. *arXiv preprint arXiv:1706.09847*, 2017.

David P. Helmbold, Nicholas Littlestone, and Philip M. Long. On-line learning with linear loss constraints. *Information and Computation*, 161(2):140 – 171, 2000a. ISSN 0890-5401. doi: http://dx.doi.org/10.1006/inco.2000.2871.

David P Helmbold, Nicholas Littlestone, and Philip M Long. Apple tasting. *Information and Computation*, 161(2):85–139, 2000b.

Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fair learning in markovian environments. *CoRR*, abs/1611.03071, 2016. URL http://arxiv.org/abs/1611.03071.

Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. Rawlsian fairness for machine learning. *arXiv preprint arXiv:1610.09559*, 2016a.

Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 325–333. Curran Associates, Inc., 2016b. URL http://papers.nips.cc/paper/6355-fairness-in-learning-classic-and-contextual-bandits.pdf.

Sampath Kannan, Michael Kearns, Jamie Morgenstern, Mallesh Pai, Aaron Roth, Rakesh Vohra, and Z Steven Wu. Fairness incentives for myopic agents. *arXiv preprint arXiv:1705.02321*, 2017.

Akshay Krishnamurthy, Alekh Agarwal, and Miro Dudik. Contextual semibandits via supervised learning oracles. In *Advances In Neural Information Processing Systems*, pages 2388–2396, 2016.

Kristian Lum and William Isaac. To predict and serve? *Significance*, pages 14 – 18, October 2016.

George O. Mohler, Martin B. Short, Sean Malinowski, Mark Johnson, George E. Tita, Andrea L. Bertozzi, and P. Jeffrey Brantingham. Randomized controlled field trials of predictive policing. *Journal of the American Statistical Association*, 110(512):1399 – 1411, 2015.

Gergely Neu and Gábor Bartók. An efficient algorithm for learning with semi-bandit feedback. In *International Conference on Algorithmic Learning Theory*, pages 234–248. Springer, 2013.

NorthPointe, Inc. Compas. http://www.northpointeinc.com/files/downloads/FAQ_Document.pdf.

Antonio Piccolboni and Christian Schindelhauer. Discrete prediction games with arbitrary feedback and loss. In *International Conference on Computational Learning Theory*, pages 208–223. Springer, 2001.